IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellant:	Wei	Patent Application	
Serial No.:	10/777,438	Group Art Unit:	2116
Filed:	February 11, 2004	Examiner:	Yanchus III, Paul B.

For: MEDTHOD AND APPARATUS FOR PROVIDING UPDATED SYSTEM LOCALITY INFORMATION DURING RUNTIME

Appeal Brief

Table of Contents

	Page
Real Party in Interest	2
Related Appeals and Interferences	3
Status of Claims	4
Status of Amendments	5
Summary of Claimed Subject Matter	6
Grounds of Rejection to be Reviewed on Appeal	8
Arguments	9
Claims Appendix	14
Evidence Appendix	18
Related Proceedings Appendix	19

Real Party in Interest

The assignee of the present invention is Hewlett-Packard Company.

Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

Status of Claims

Claims 1-20 remain pending. Claims 1-20 have been rejected. This appeal involves Claims 1-20.

Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

Summary of Claimed Subject Matter

Independent Claim 1 recites "A method (400) for providing updated system locality information (SLI) during runtime," which is described according to various embodiments at line 28 on page 10 to line 15 on page 14 and Figure 4. "Collecting (402) system locality information (300) at boot time to be provided to an operating system (510), said system locality information (300) describing distances between devices within an integrated processing system," is described according to various embodiments at lines 1-17 on page 11 and Figures 3-5. "Notifying (406) said operating system (510) that a triggering event has occurred, wherein said triggering event potentially alters said system locality information," which is described according to various embodiments at line 25 on page 11 to line 12 on page 12 and Figures 4 and 5. "Providing (410, 408, 412) said updated system locality information during said runtime to said operating system (510) upon a request from said operating system (510), said updated system locality information reflecting said distances between said devices within said integrated processing system after occurrence of said triggering event," is described according to various embodiments at line 14 on page 12 to line 15 on page 14 and Figures 4 and 5.

Independent Claim 18 recites, "A computer program embodied on a computer readable medium for providing dynamically updated system locality information," which according to various embodiments is described at lines 1-17 on page 11 and lines 11-14 on page 14. "Creating (402) a system locality information table (300), said system locality information table (300) being populated with boot time system locality information, wherein said system locality information table (300) describes distances between devices within an integrated processing system," is described according to various embodiments at lines 1-17 on page 11. "Updating said system locality information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said distances between said devices within said integrated

processing system," which is described according to various embodiments at lines 11-14 on page 14.

Independent Claim 20 recites "An apparatus (500) for updating system locality information," which according to various embodiments is described at lines 16 on page 15 to line 13 on page 16 and Figure 5. "A system locality information table (SLIT) creator (502) for creating a SLIT (300) coupled to an operating system (510), said SLIT (300) being populated with boot time system locality information, wherein said system locality information describes distances between devices within an integrated processing system," is described according to various embodiments at line 29 on page 15 to line 3 on page 16 and Figures 3-5. "A triggering event detector (506) coupled to said operating system (510), said triggering event detector (506) capable of detecting (404) an occurrence of a triggering event, where said triggering event may potentially alter said distances between said devices within an integrated processing system," is described according to various embodiments at line 21 on page 14 to line 14 on page 15 and Figures 4-5. "A SLIT updator (504) coupled to said operating system (510) and further coupled to said triggering event detector (506), wherein, upon a receipt of a notification of an occurrence of said triggering event from said triggering event detector (506), said SLIT updator (504) provides (410, 408, 412) said updated system locality information to said operating system (510) based on said altered distances between said devices of said integrated processing system," is described according to various embodiments at line 16 on page 15 to line 13 on page 16, line 14 on page 12 to line 15 on page 14 and Figures 4-5.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1-20 are rejected under 35 U.S.C. §103(a) as being anticipated by U.S. Patent Application Publication No. 2004/0243534 by Cutler et al. (referred to hereinafter as "Cutler") in view of Hewlett-Packard, ACPI System Locality Information Table Interface Version 0.9 (referred to hereinafter as "ACPI Asserted Art").

Arguments

1. Whether Claims 1-20 are unpatentable under 35 U.S.C. 103(a) over Cutler in view of ACPI Asserted Art

Applicant has reviewed the asserted art and respectfully submits that the embodiments recited by the instant application serial no. 10/777,438 are neither taught nor suggested by Cutler or ACPI Asserted Art, alone or in combination, for at least the following rationale.

Applicant respectfully submits that "[i]t is improper to combine references where the references teach away from their combination" (emphasis added; MPEP 2145(X)(D)(2); *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983)). Applicant respectfully notes that "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention" (emphasis in original; MPEP 2141.02(VI); *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984)). Further, Applicant respectfully submits that, "[w]ith regard to rejections under 35 U.S.C. 103, the examiner must provide evidence which as a whole shows that the legal determination sought to be proved (i.e., the reference teachings establish a *prima facie* case of obviousness) is more probable than not" (emphasis added) (MPEP 2142).

In particular, "if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious" (emphasis added) (MPEP 2143.01(VI); *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959)). Further, "[i]f the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed amendment" (emphasis added) (MPEP 2143.01(V); *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)).

More specifically, Applicant respectfully submits that there is no motivation to combine the teachings of Cutler and ACPI Asserted art, because various combinations of the asserted art teach away from the suggested modification. For example, Applicant understands the combination of Cutler and ACPI Asserted art to change each other's principles of operation, or to render each others' teachings unsatisfactory for their intended purposes, or a combination thereof as will be described in more detail.

CUTLER

This section describes Applicant's understanding of what Cutler teaches. Applicant understands Cutler to teach autonomously and dynamically generating an ACPI table entry that includes any added AML code therein for supporting a hardware element C that has been added (0039). An agent 207 accesses a database 206 and utilizes device-specific properties stored in the database 206 for hardware element C to generate any AML code that supports hardware element C (0039). The composite pattern of information stored in the database 206, 605 resembles a tree, and therefore the database 206, 605 is referred to as a "Device Tree" (0074).

ACPI ASSERTED ART

This section describes Applicant's understanding of what the ACPI Asserted Art teaches. The ACPI Asserted Art states in the fourth and fifth paragraphs under the second section entitled "System Locality Information Table Definition,"

> The SLIT can be viewed as a matrix of distances, with row I of the matrix indicating the distance from locality I to every locality (including itself). Each table entry is a 1-byte unsigned integer. To get the distance from locality i to locality j, read the i*(Localities)+j entry in the matrix. Except for the distances from the locality to itself, each distance is stored twice in the matrix.
>
> What this means is that the diagonal elements of the matrix, the distances from the locality to itself, which are the SMP distances, are all given a

value of 10. The distances for the non-diagonal elements are scaled to be relative to the SMP distance, so, for example, if the distance from locality i to locality j is 2.4 times the SMP distance, a value of 24 would be stored in table entry i*(localities)+j and in j*(localities)+i (emphasis added).

THE COMBINATION OF CUTLER AND THE ACPI ASSERTED ART

This section describes why Applicant believes that Cutler and ACPI Asserted art teach away from each other and therefore they cannot be combined to render the embodiments recited by the instant application obvious. Cutler uses a "Device Tree." ACPI Asserted art uses a matrix. Modifying Cutler to use a matrix instead of a device tree or modifying ACPI Asserted art to use a device tree instead of a matrix would change their principles of operation and would render them unsatisfactory for their intended purposes.

RESPONSE TO ARGUMENTS

This section further describes Appellant's understanding of why Cutler and ACPI asserted art cannot be combined to render embodiments of the instant application obvious. The Office Action asserts in the response to arguments section on pages 7 and 8 that Cutler's tree and ACPI asserted art's matrix do not teach away from each other.

Appellant respectfully disagrees. The ACPI asserted art's System Locality Information Table, which is a matrix, is the data structure that the ACPI asserted art uses in order to achieve ACPI asserted art's intended purpose of processing "…the locality information to increase its performance on a NUMA system" (last two lines on page 3 of the ACPI asserted art). The use of the matrix is essential to achieving ACPI's intended purpose because using a matrix data structure results in being able to determine distances between processors, memory controllers, and host bridges (lines 1-3 of section 2) in order to increase performance on a NUMA system.

Cutler's composite pattern of information, which resembles a tree, is the data structure that Cutler uses in order to achieve Cutler's intended purpose of

being capable adding a hardware element C to a system 100 during the system's run time (lines 4-9 of paragraph 0034). A device tree is also referred to as a Banyan framework (0074). The use of a device tree is essential to achieving Cutler's intended purpose because "As the system hardware itself is built of aggregations of aggregated elements, database 605 in Banyan explicitly mirrors these aggregations with the composite software design patterns" (0074).

Therefore, Cutler's tree is a data structure for achieving Cutler's intended purpose. ACPI asserted art's matrix is a data structure for achieving Cutler's intended purpose. Modifying ACPI asserted art to use a tree would change ACPI asserted arts principle of operation and would render the ACPI asserted art inoperable for ACPI asserted art's intended purpose because a tree could not be used to determine distances between localities represented by various entries in the tree. Similarly, modifying Cutler to use a matrix would change Cutler's principle of operation and would render Cutler inoperable for Cutler's intended purpose because Appellant does not understand a matrix being capable of explicitly mirroring aggregations with the composite software designs patterns.

SUMMARY

For at least the reasons that Applicant believes that Cutler and ACPI Asserted art teach away from each other, Applicant respectfully submits that Cutler and the ACPI asserted art cannot be combined to render the embodiments recited by the instant application serial no. 10/777,438 obvious.

In summary, the Applicant respectfully requests that the Board reverse the rejections of Claims 1-20.

The Applicant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Applicant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,

WAGNER BLECHER LLP

Date: 06/03/2009

/John P. Wagner, Jr./

John P. Wagner, Jr.

Registration Number: 35,398

WAGNER BLECHER LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076
408-377-0500

Claims Appendix

1. A method for providing updated system locality information (SLI) during runtime comprising:

collecting system locality information at boot time to be provided to an operating system, said system locality information describing distances between devices within an integrated processing system;

notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said system locality information; and

providing said updated system locality information during said runtime to said operating system upon a request from said operating system, said updated system locality information reflecting said distances between said devices within said integrated processing system after occurrence of said triggering event.

2. The method of claim 1 further comprising:

creating a system locality information table (SLIT), wherein said SLIT is populated with device distances collected at boot time.

3. The method of claim 2 wherein said notification of said operating system of an addition causes said operating system to invoke a process for updating said SLIT.

4. The method of claim 3 wherein data for said update of said SLIT is provided to said operating system upon an invocation of an Advanced Configuration and Power Interface (ACPI) object.

5. The method of claim 4 wherein said ACPI object is _SLI.

6. The method of claim 1 wherein said triggering event is based on an online addition of a device.

7. The method of claim 6 wherein said notification of said online addition is through a Bus Check notification.

8. The method of claim 1 wherein the triggering event is based on an online deletion of a processor device.

9. The method of claim 8 wherein said notification of said online deletion is through an Eject Request notification.

10. The method of claim 1 wherein the triggering event is based on an online reconfiguration of said integrated processing system, wherein said online reconfiguration affects distances between devices within said integrated processing system.

11. The method of claim 10 wherein said notification of said online reconfiguration is via a device that affected said distances.

12. The method of claim 11 wherein said notification of said online reconfiguration uses an ACPI object.

13. The method of claim 12 wherein said object is SLI Update.

14. The method of claim 3 wherein said update of said SLIT is by recreating a new table.

15. The method of claim 3 wherein said update of said SLIT is by augmenting and populating augmented cells with new system locality information.

16. The method of claim 3 wherein said update of said SLIT is by reducing and populating said SLIT with new system locality information.

17. The method of claim 3 wherein said update of said SLIT is by updating existing cells within said SLIT with new system locality information.

18. A computer program embodied on a computer readable medium for providing dynamically updated system locality information, the computer program causing a computer to perform the steps of:

creating a system locality information table, said system locality information table being populated with boot time system locality information, wherein said system locality information table describes distances between devices within an integrated processing system; and

updating said system locality information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said distances between said devices within said integrated processing system.

19. The computer program of claim 18 wherein said computer program further causes said computer to:

invoke a bus check notification upon an online addition of a device, wherein said bus check notification indicates to said operating system that a re-enumeration of a device tree needs to be performed, and wherein said operating system invokes a _SLI procedure that returns updated system locality information resulting from said online addition;

invoke an Eject Request notification upon an online deletion of a device, wherein said Eject Request notification indicates to said operating system that a re-enumeration of a device tree needs to be performed, and wherein said operating system invokes a _SLI procedure that returns updated system locality information resulting from said online deletion; and

invoke an SLI Update notification upon an online reconfiguration of said integrated processing system, wherein said SLI Update notification indicates to said operating system that a re-enumeration of a device tree needs to be performed, and wherein said operating system invokes a _SLI procedure associated with a device sending said SLI Update notification that returns updated system locality information resulting from said online reconfiguration.

20. An apparatus for updating system locality information comprising:

a system locality information table (SLIT) creator for creating a SLIT coupled to an operating system, said SLIT being populated with boot time system locality information, wherein said system locality information describes distances between devices within an integrated processing system;

a triggering event detector coupled to said operating system, said triggering event detector capable of detecting an occurrence of a triggering event, where said triggering event may potentially alter said distances between said devices within an integrated processing system; and

a SLIT updator coupled to said operating system and further coupled to said triggering event detector, wherein, upon a receipt of a notification of an occurrence of said triggering event from said triggering event detector, said SLIT updator provides said updated system locality information to said operating system based on said altered distances between said devices of said integrated processing system.

Evidence Appendix

None

Related Proceedings Appendix

None